Commonly used software to generate HDRs and pitfalls

DONG HYUN KIM

23. International Radiance Workshop, 29.8.2025, Lausanne (CH)

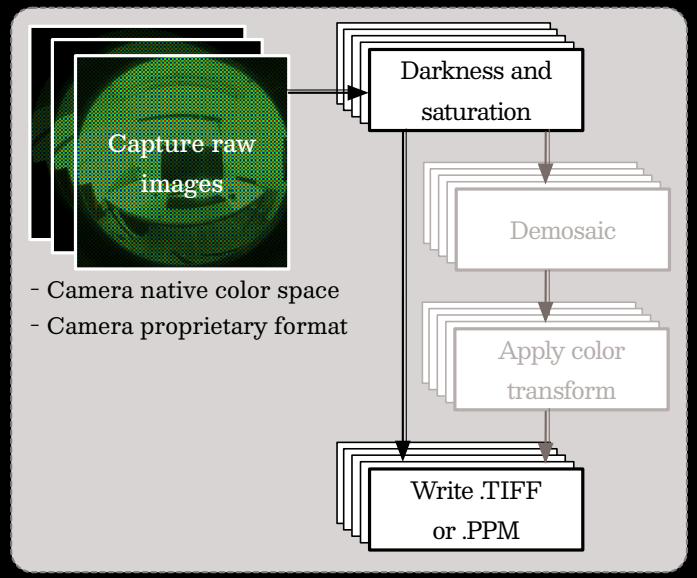
kim@arch.ethz.ch

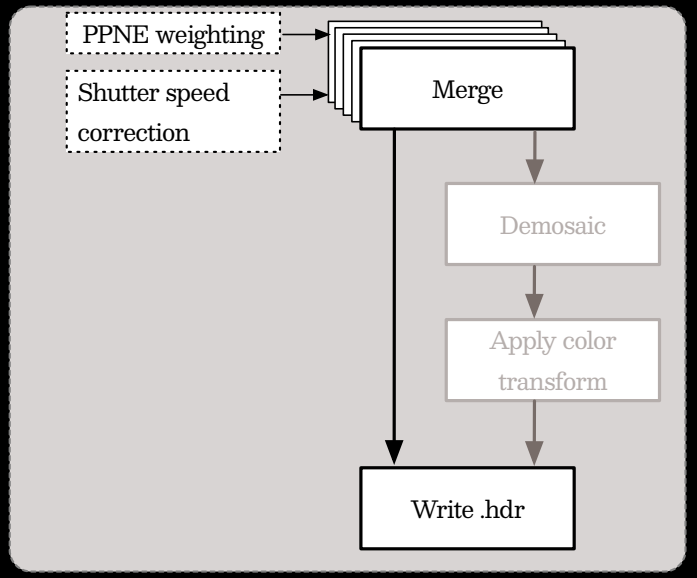
A / S Architecture and Building Systems

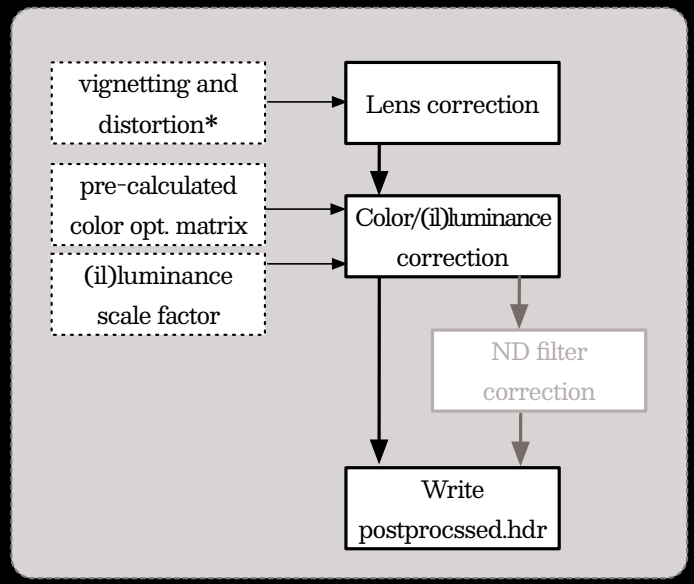
{HiDyn}

ETH zürich

Simplified HDR generation sequence using a commercial-grade digital camera





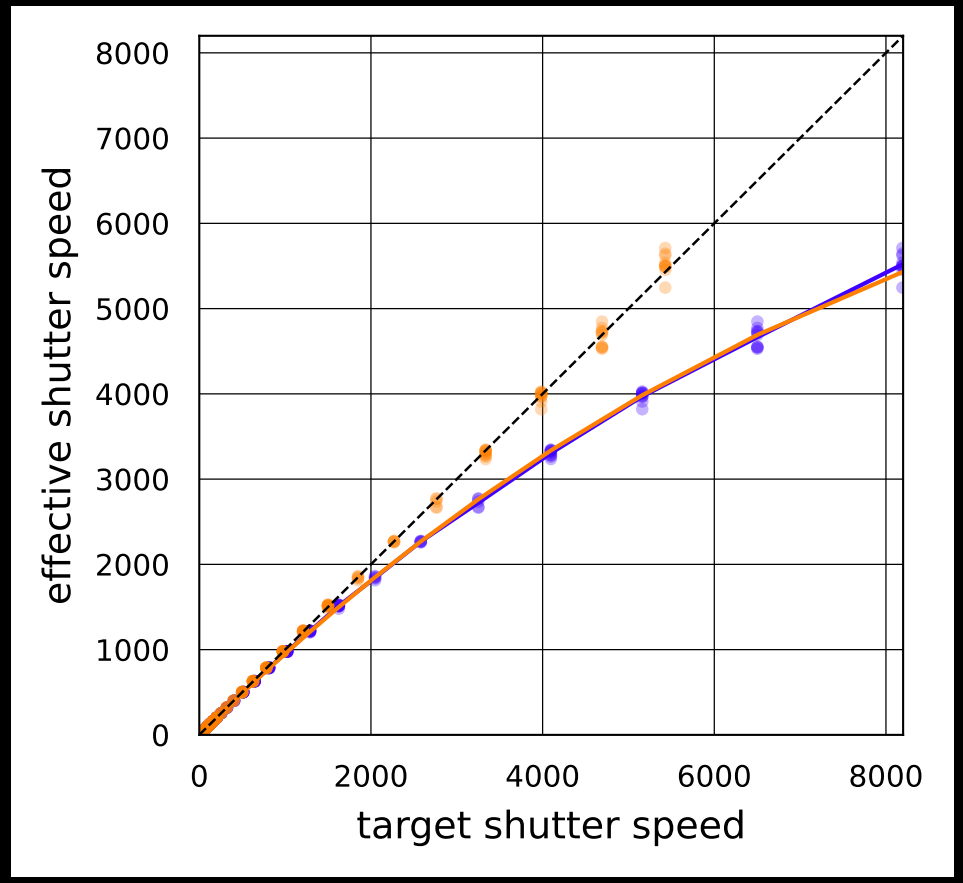


- **RAW processing software** handles this stage; banding/wrong colors in regions with saturated exposures are typically caused by mis-set saturation values.
- **Color transformation** converts camera native color space to device-independent coordinates (e.g., CIE XYZ, sRGB) and can be applied either before merging LDR frames or after HDR synthesis.

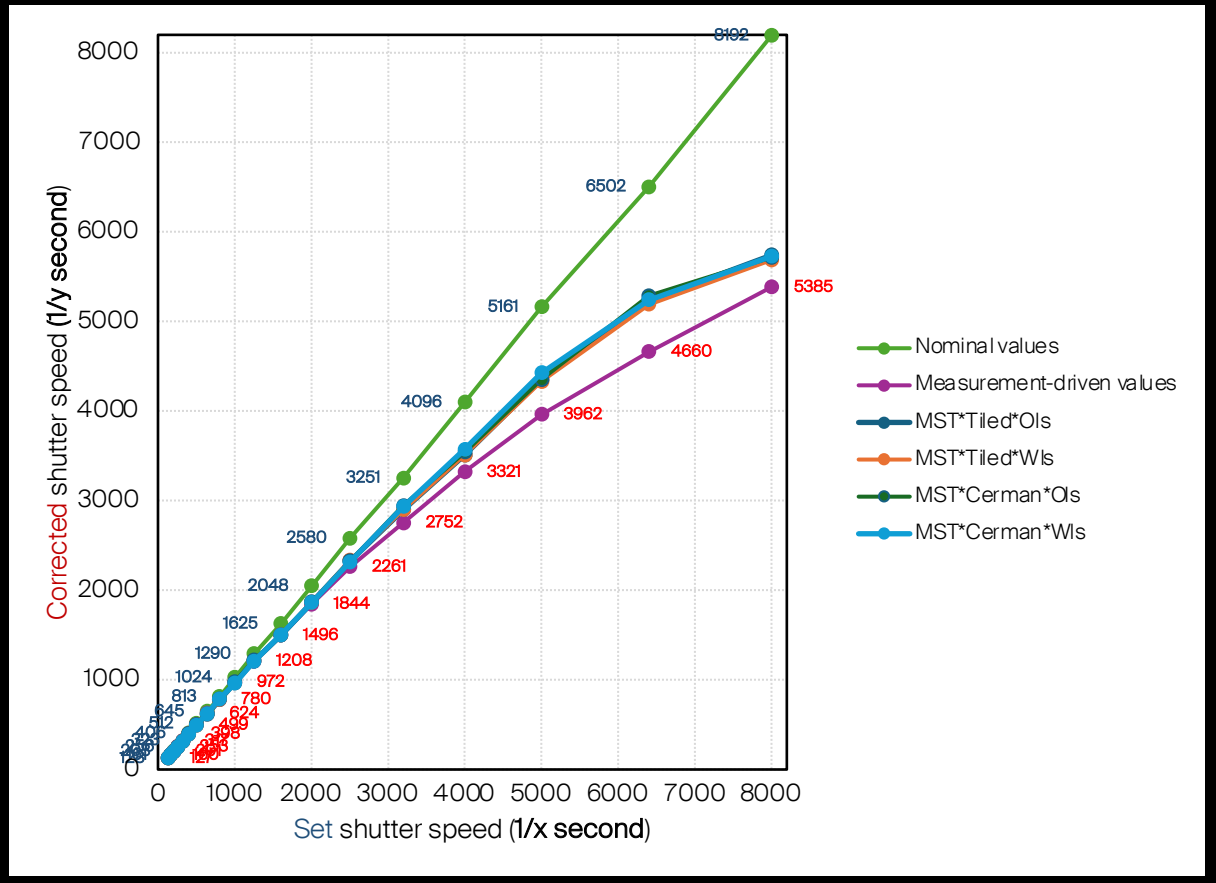
- **Shutter speed correction** may be required when the effective exposure time deviates significantly from the target (nominal) value.

- **Photometric adjustment (or calibration)** is often done by a measuring spot luminance on a gray card or vertical illuminance, then scaling all pixel luminances by a single global factor.
- Several **color calibration** methods have been proposed to map pixel values to correct colorimetry, via chart-based regression or spectral models.

Shutter speed (exposure) correction

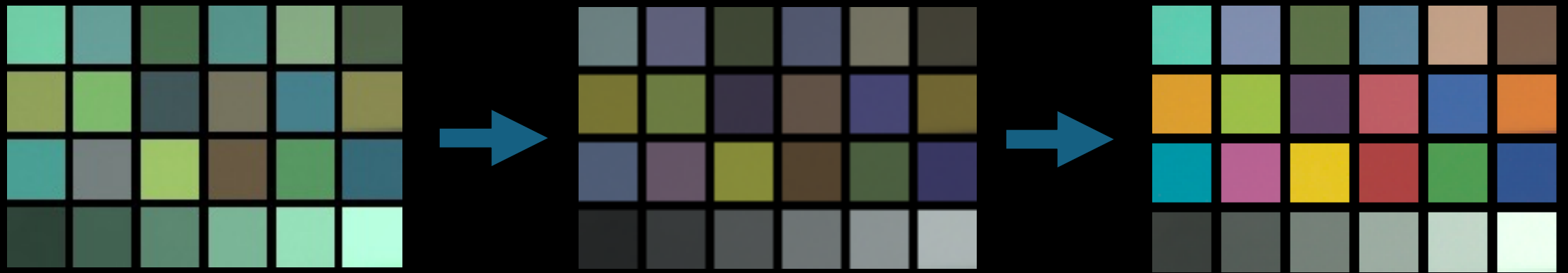


Shutter-speed deviations reported by Wasilewski et al. (Canon 70D, mechanical shutter)



Shutter-speed deviations identified using various Hanji P & Manituk R (2023) method (Canon 70D, mechanical shutter)

Color transformation (or calibration)



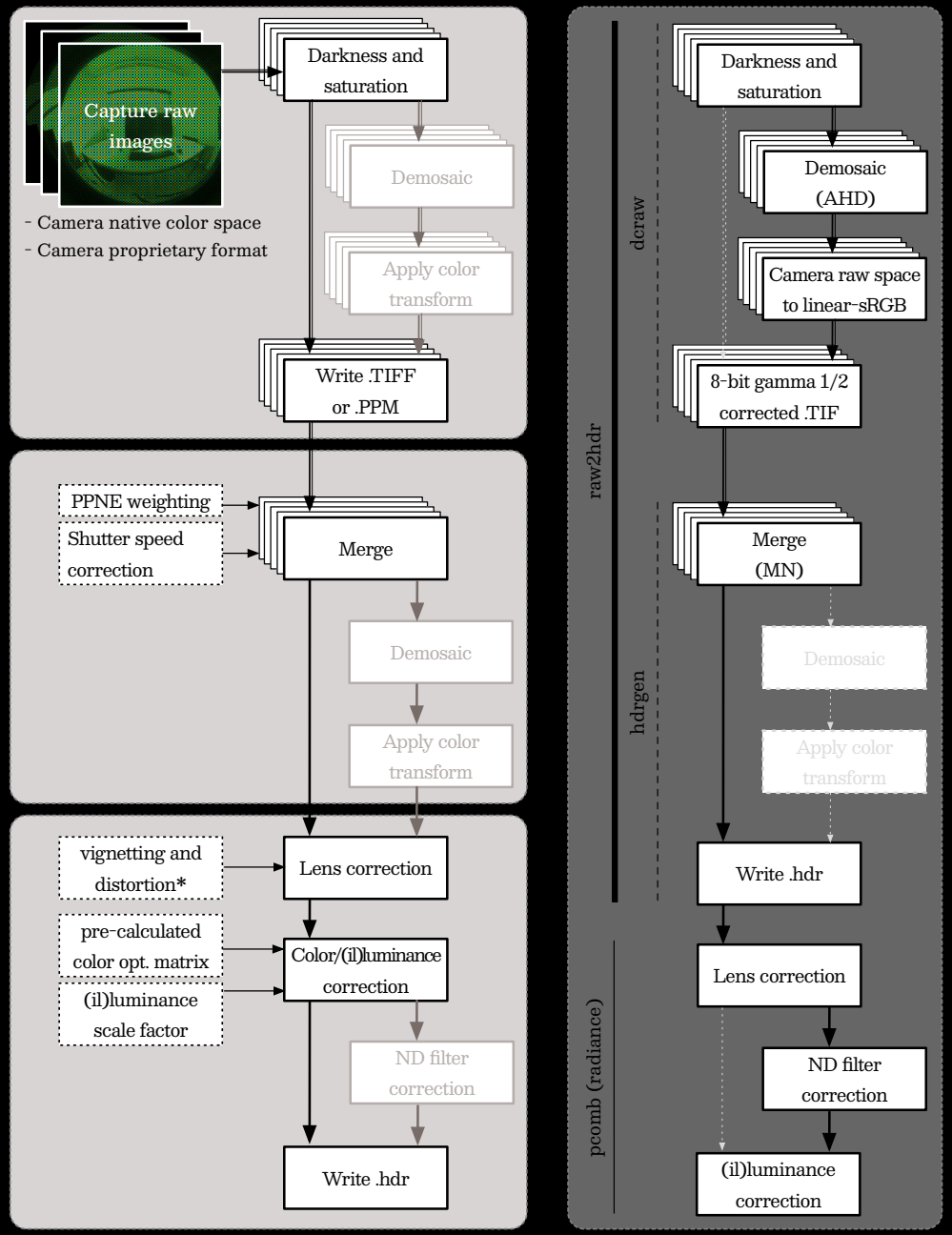
Camera native color space

CIE XYZ

sRGB or Adobe RGB or so..

- **Color calibration** (also called **color optimization**) with a ColorChecker is specified in ISO 17321-1, Method B. Given reference CIE XYZ values and corresponding camera native RGBs, first estimate a 3×3 linear transformation by least squares, then refine it with a non-linear optimization that minimizes CIE Lab* color-difference (ΔE) to a local minimum.
- Several calibration methods have been proposed in the literature, including those by Cauwerts et al. (2019), Jung and Inanici (2019), Yamaguchi et al. (2019), Varghese et al. (2014), Kim and Kautz (2008), and Wasilewski et al. What are the differences between these methods?
- The literature suggests that color calibration should be **robust across a wide range of broad-spectrum conditions**. Does this imply that once color calibration is performed, **photometric calibration is no longer necessary**?

Simplified HDR generation process



raw2hdr

How is raw2hdr used by default to generate HDR images?

```
raw2hdr –o output.hdr *.CR2
  dcraw –o 1 –W –g 2 0 –T *.CR2
    dcraw_emu –t 0 –W –g 2 0 –o 1  –T –b 1
    dcraw_emu –t 0 –W –g 2 0 –o 0  –T –b 1
  hdrgen –e –a –r sqr.rsp –o output.hdr –c sRGB *.tif
```

Consider replacing *dcraw* with *libraw* using this command line.

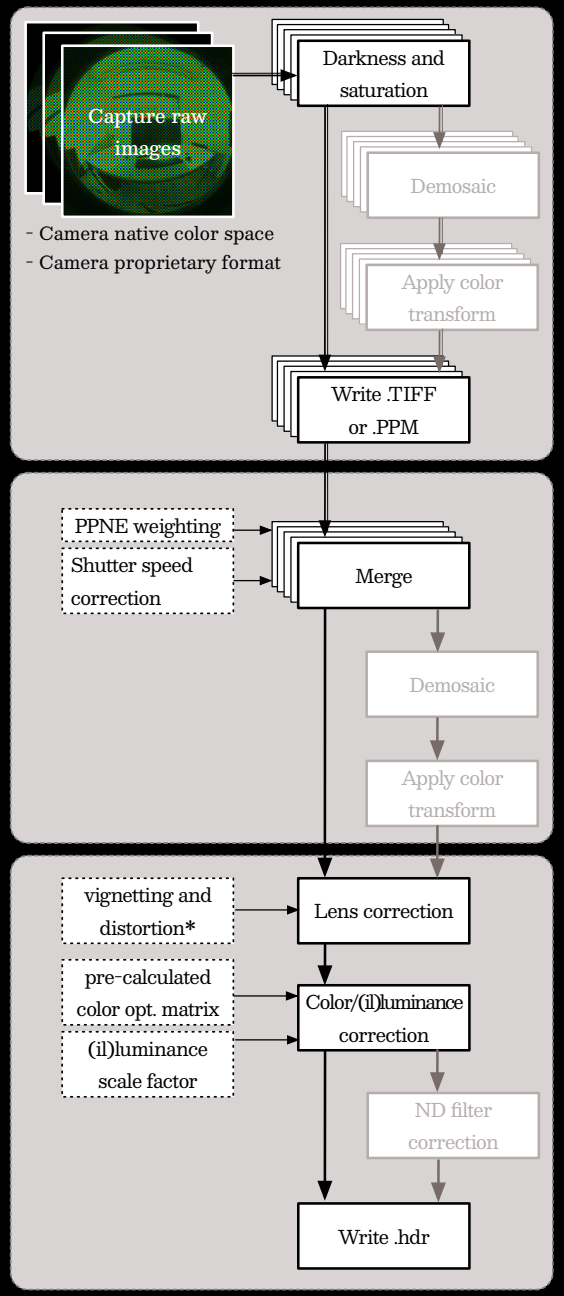
If your camera model is not yet supported by *libraw* or prefer to retain the camera's native color space after conversion.

The scale factor should be approximately 1. If it significantly deviates from 1 (e.g., 2 or higher), consider deriving the scale factor using the measured vertical illuminance value.

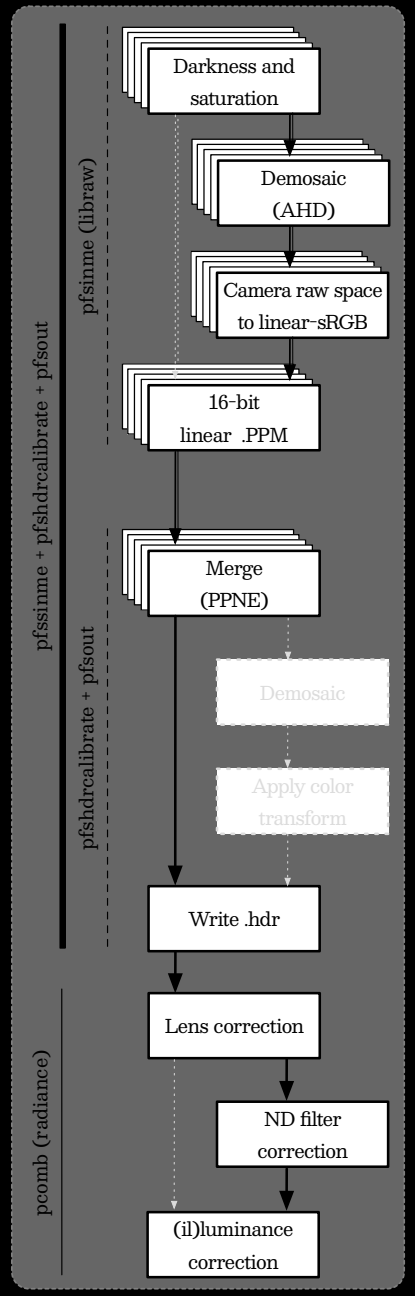
The post-processing part is well explained in the existing literature.

```
pcomb –f fisheye_corr.cal output.hdr > output_equidistant.hdr
pcomb –f ND_correction.cal output.hdr > output_ndcorrected.hdr
pcomb –s scale_factor output.hdr > output_luminancescaled.hdr
```

Simplified HDR generation process



pfstools



How is pfstools used by default to generate HDR images?

```
pfsinme *.CR2 | pfshdrcalibrate –r linear –c none | pfsout output.hdr
```

pfsinme *.CR2

- dcraw –c –o 1 -4 **-w**
- dcraw_emu -o 1 -4 **–w**
- dcraw_emu –t 0 –W –o 0 –T –b 1

dcraw is used in pfstools v2.1 or earlier

libraw is used in pfstools v.2.2

If your camera model is not yet supported by libraw or you prefer to retain the camera's native color space after conversion.

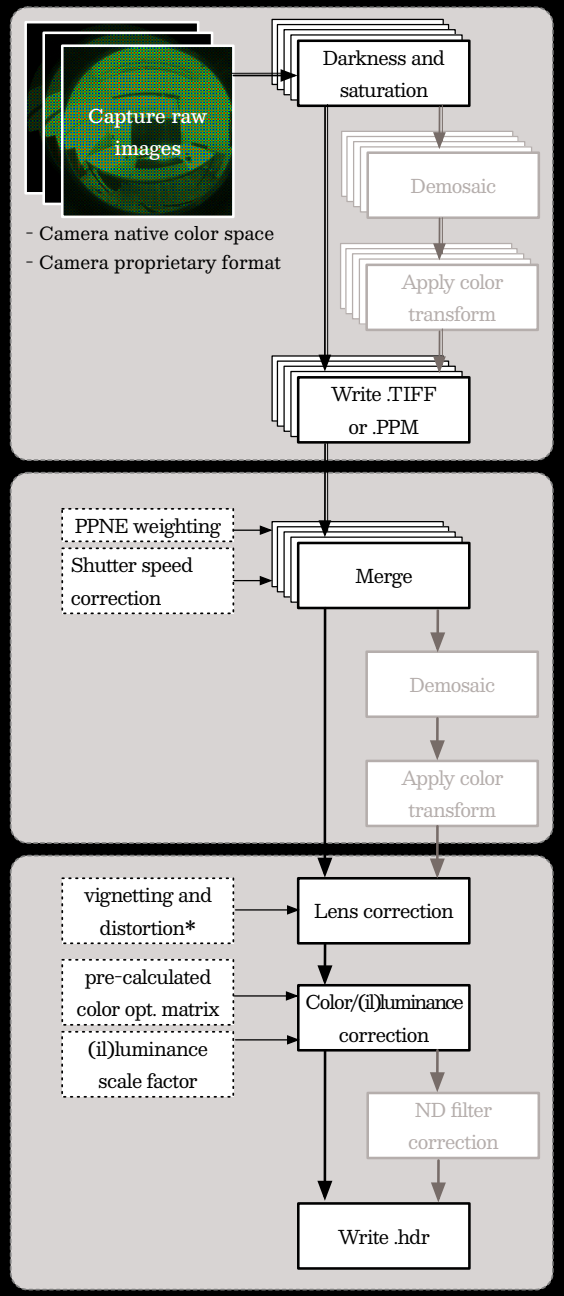
pfshdrcalibrate –r linear –c none

pfsout output.hdr

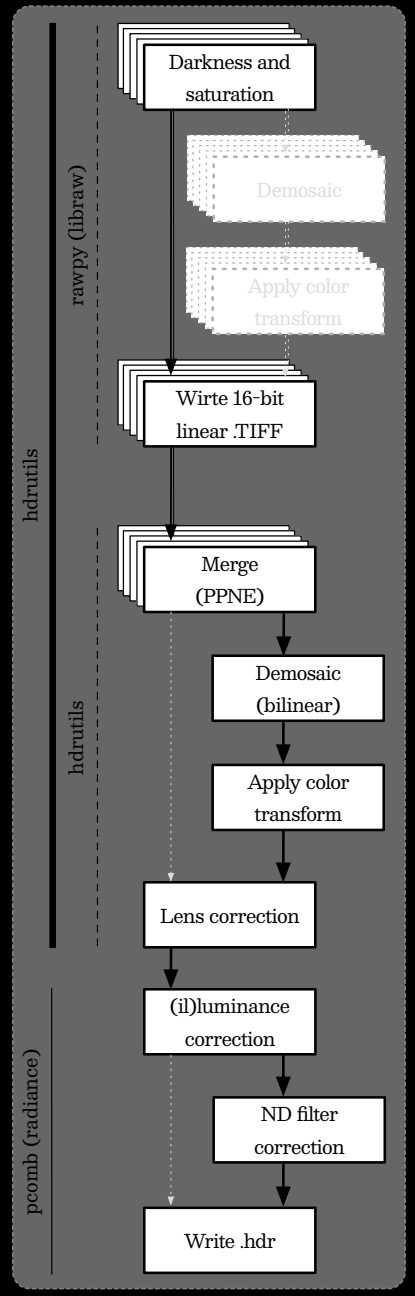
By default, pfstools includes **a normalization process in HDR merging**, and therefore the scale factor can be very large (e.g., k>10,000 or >10,000,000).

```
pcomb –f fisheye_corr.cal output.hdr > output_equidistant.hdr
pcomb –f ND_correction.cal output.hdr > output_ndcorrected.hdr
pcomb –s scale_factor output.hdr > output_luminancescaled.hdr
```

Simplified HDR generation process



HDRutils



How is HDRutils used to by default generate HDR images?

Example usage with no shutter correction

```
hdrmerge --merge-first --color-space sRGB --demosaic bilinear --return-exif-exp -bits 32 -black-level 2048 -clip-highlights -saturation-percent 0.7 -o output.hdr *.CR2
```

Example usage with shutter correction

```
hdrmerge --merge-first --color-space sRGB --demosaic bilinear --return-exif-exp --estimate-exp MST --outlier tiled -solver OLS --bits 32 --black-level 2048 --clip-highlights --saturation-percent 0.7 -o output.hdr *.CR2
```

rawpy no_auto_bright output_bps=16, user_wb=wb, user_flip=0, output_color=sRGB

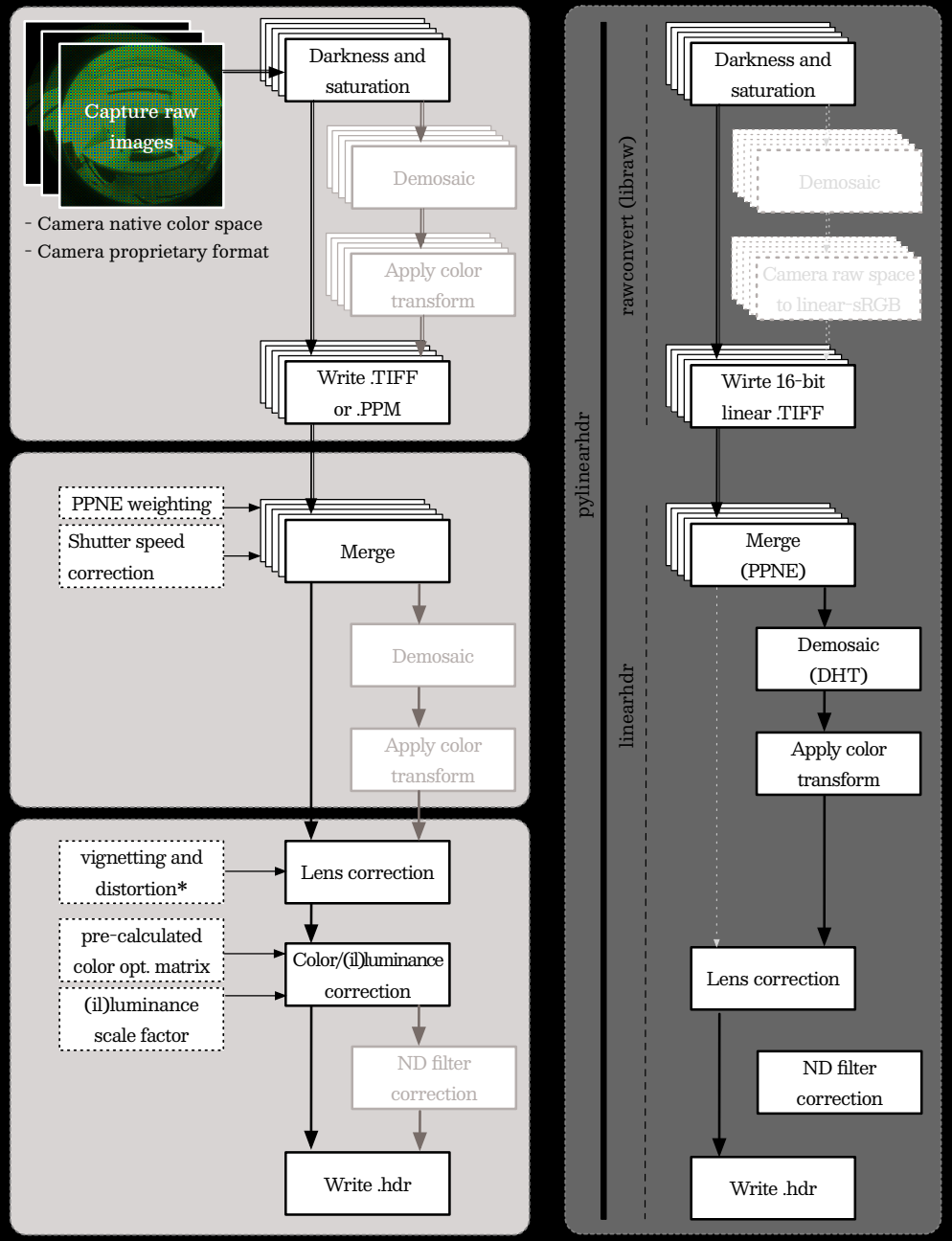
mergy.py ←

HDRutils offers flexibility, allowing users to choose **different demosaicing algorithms** and **color transforms before or after merging**. Several algorithms are included to **perform robust estimation of effective shutter values** (shutter calibration).

By default, HDRutils includes kind of **a normalization process in HDR merging**, and therefore the scale factor can be varying.

```
pcomb -f fisheye_corr.cal output.hdr > output_equidistant.hdr

pcomb -f ND_correction.cal output.hdr > output_ndcorrected.hdr

pcomb -s scale_factor output.hdr > output_luminancescaled.hdr
```

Simplified HDR generation process



pylinearhdr

How is pylinearhdr used to generate HDR images?

Example usage with exposure correction and color calibration

```
pylinearhdr run –black 2049 –white 11000 –shutterc (shutter_calibration_result) –xyzcam (color_calibration_result) –fo (aperture_calibration_result) –colorspace srgb "*.CR2" > output.hdr
```

Example usage with no correction or calibration

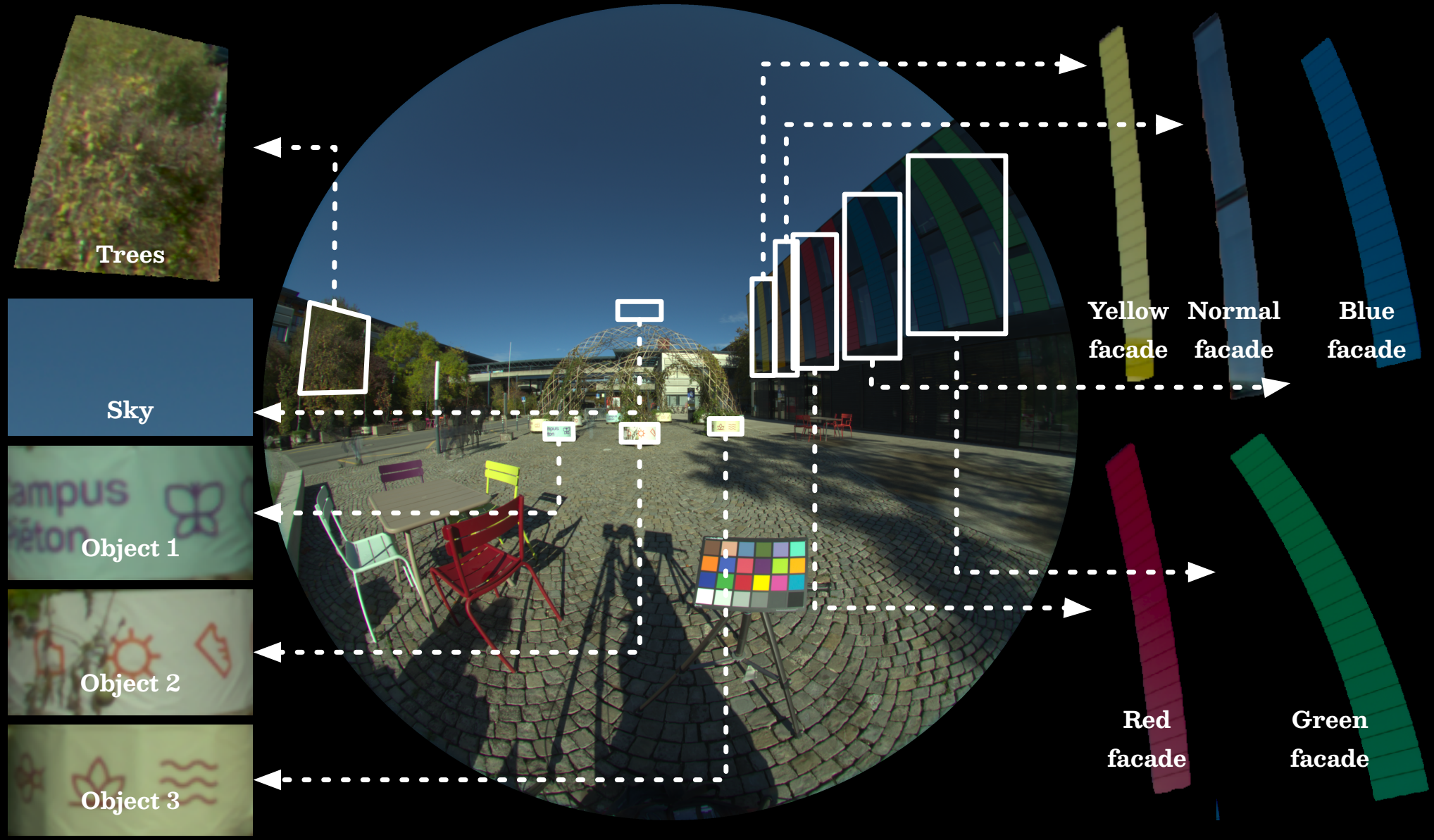
```
pylinearhdr run –black 2049 –white 11000 –colorspace srgb "*.CR2" > output.hdr
```

pylinearhdr provides flexibility in the HDR merging workflow, offering options such as **different demosaicing algorithms**, **color transforms before or after merging**, and **various weighting options for merging**, such as best exposure selection or weighted averaging. It also supports shutter **speed calibration** and **color calibration**. Once color calibration is performed, **additional photometric adjustments are generally not needed across a wide spectrum**. Without color calibration, the scale factor should be approximately 1, as with raw2hdr. Further post-processing steps, such as lens projection correction and vignetting correction, are also performed in pylinearhdr.

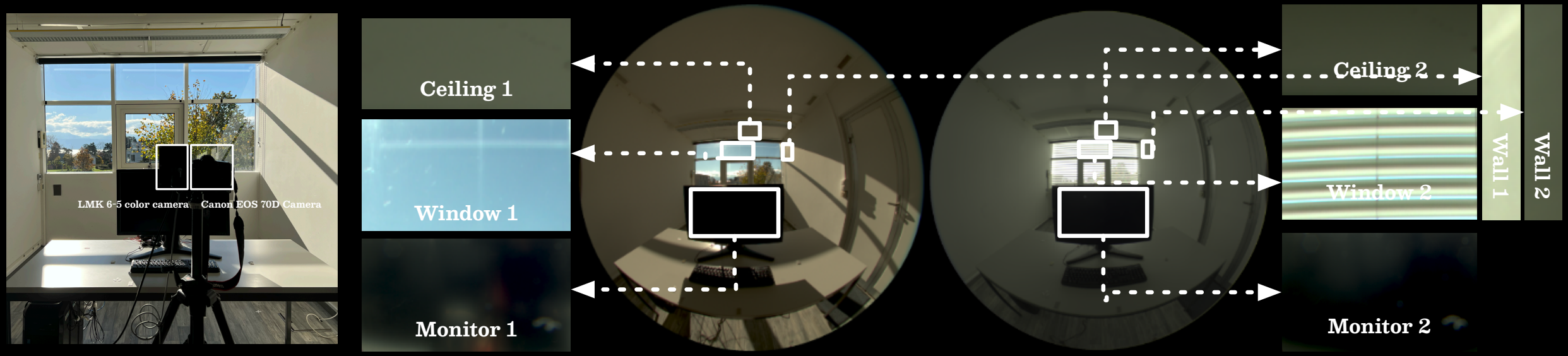
pfstools by default includes normalization process in HDR merging, and therefore scale factor can be very large (e.g., >10,000 or 10,000,000).

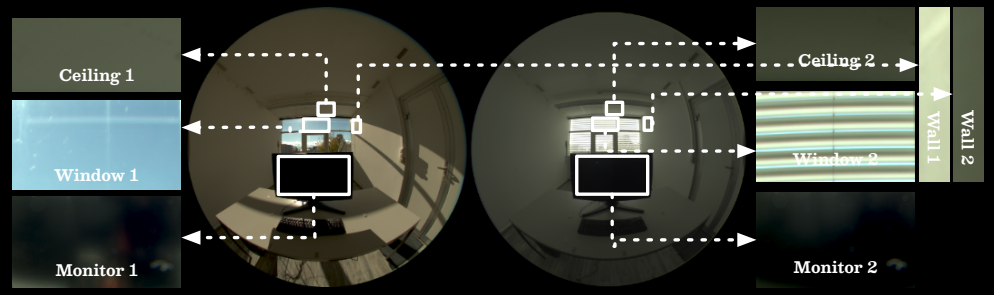
```
pfsout output.hdr

pcomb –f fisheye_corr.cal output.hdr > output_equidistant.hdr

pcomb –f ND_correction.cal output.hdr > output_ndcorrected.hdr

pcomb –s scale_factor output.hdr > output_luminancescaled.hdr
```

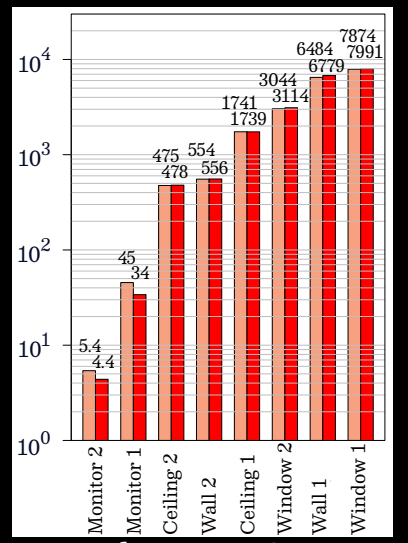
Performance evaluation of different HDR generation software based on real-world captured scenes



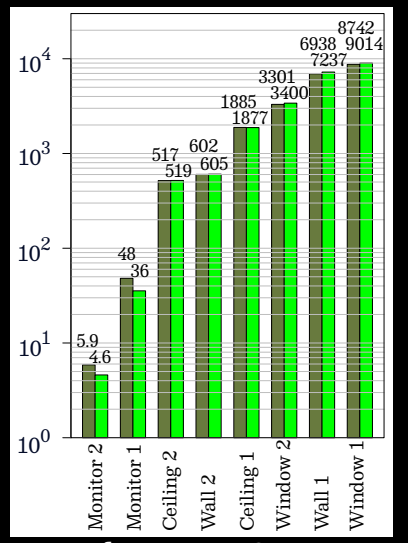
Performance evaluation of different HDR generation software based on real-world captured indoor scenes



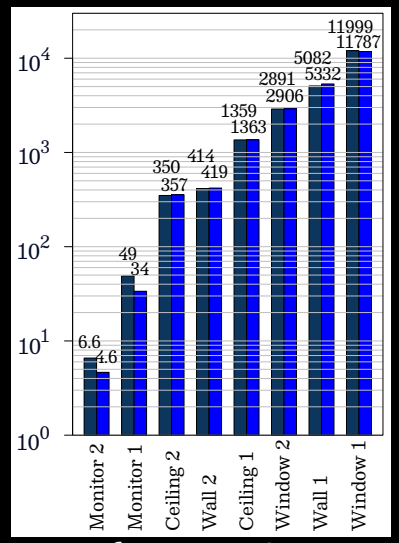




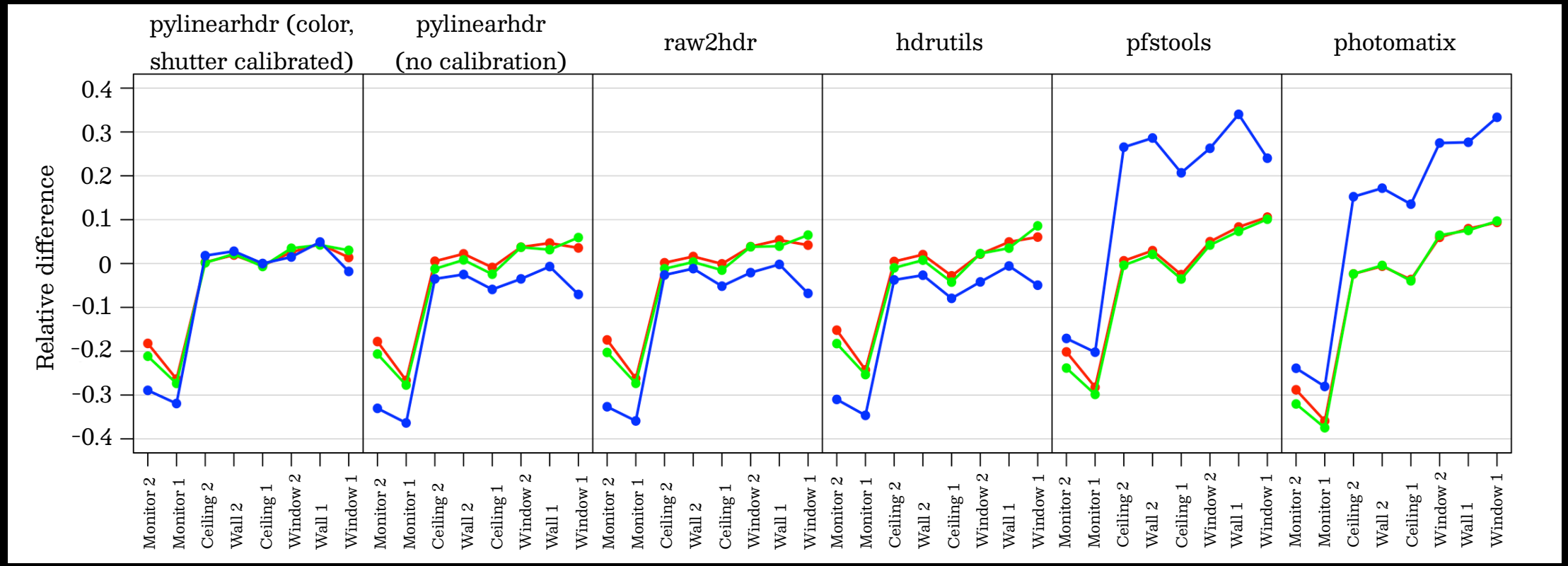
Reference CIE X

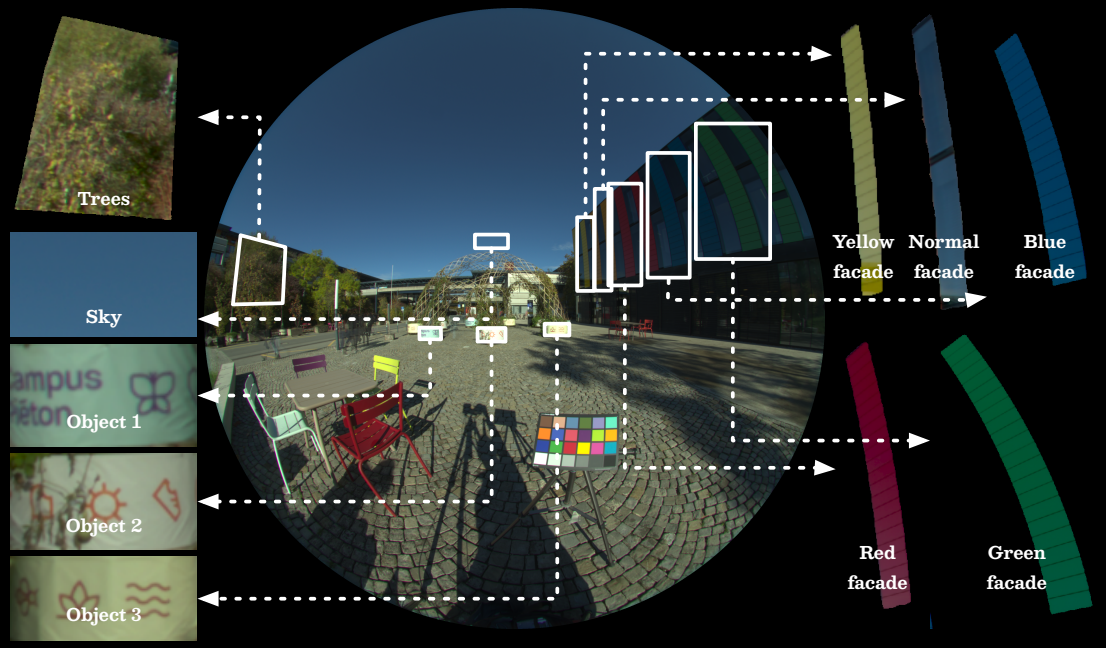


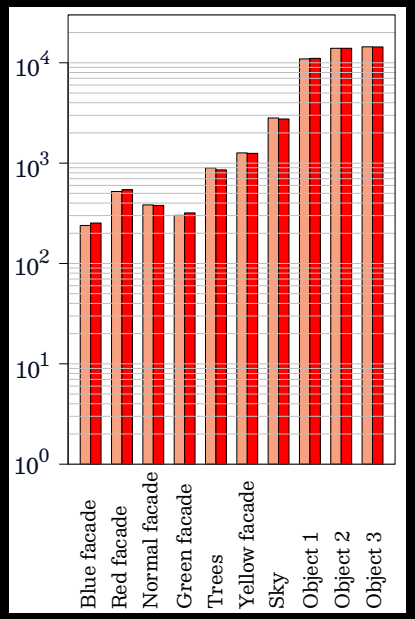
Reference CIE Y



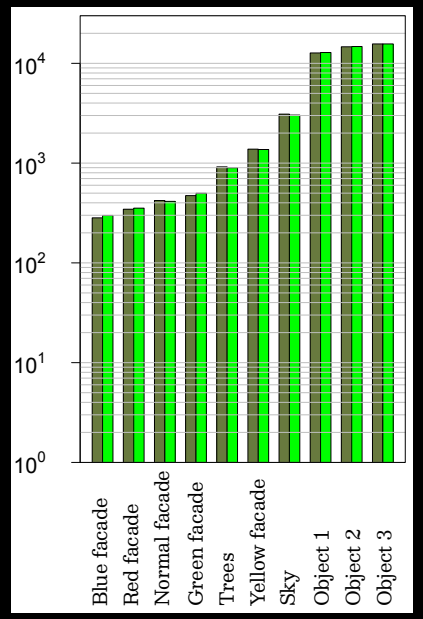
Reference CIE Z



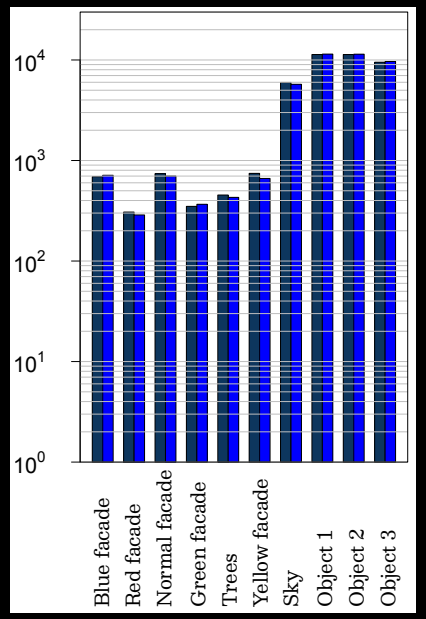




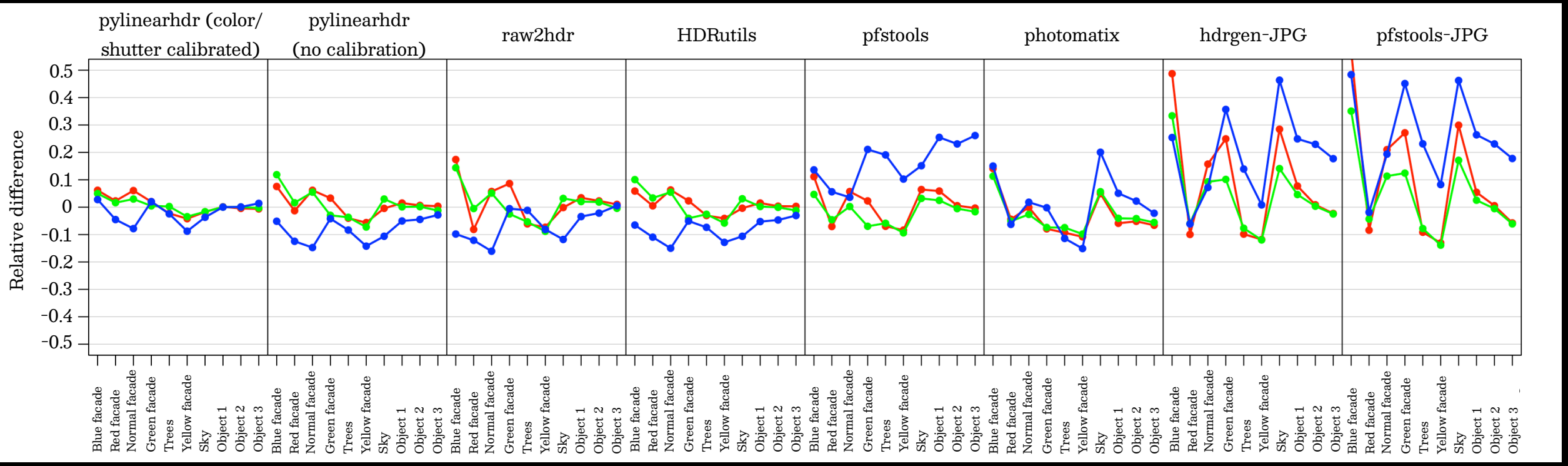
Reference CIE X



Reference CIE Y



Reference CIE Z



Conclusion

- **RAW processing variability**: Different HDR generation software employs varied RAW processing methods, with white balance settings in the preparation stage affecting the accuracy of converted pixel values relative to real-world colorimetric values.
- **Performance similarity**: Within the presented dataset, pylinearhdr, raw2hdr and HDRutils show similar data patterns.
- **Color and shutter calibration**: HDR images generated by pylinearhdr with color and shutter speed calibration show reduced relative differences compare to reference values, performing more closely to expectations than other software.
- **JPEG-based HDR merging**: HDR files created from JPEG images (using hdrgen and/or pfstools) generally exhibit higher relative differences in colorimetric values both indoor and outdoor scenes.
- **Commercial proprietary software**: Photomatix, a proprietary software shows greater relative differences in colorimetric values compared to other RAW-based HDR merging software and the source of these differences cannot be traced due to the proprietary nature of the software.